IFW/AF

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Mail Stop Appeal Brief-Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on November 9, 2005.

Robert E. Malm

In re Application of:

CUREY et al.

Serial Number: 09/821,537

Filing Date: 03/28/2001

For: PARTITIONED EXECUTIVE STRUCTURE FOR REAL-TIME PROGRAMS

Group Art Unit: 2152

Examiner: NABIL EL-HADY

Telephone: (703) 305-8646

AMENDMENT TO REVISED SUPPLMENTAL APPEAL BRIEF

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Responding to Notification of Non-Compliant Appeal Brief, applicants request that the attached sections be substituted for the corresponding sections in subject appeal brief.

In accordance with MPEP § 1205.03(B), a new section entitled "Summary of Claimed Subject Matter" is attached containing concise explanations of the subject matter defined in the independent claims 1 and 26.

In accordance with MPEP § 1205.03 and minor non-compliances in an appeal brief, a new section entitled "Grounds of Rejection to be Reviewed on Appeal" is attached which includes "Grouping of Claims" as a subsection. A revised Table of Contents is attached which is consistent with these changes.

I believe these submissions remedy the non-compliance issues raised by the examiner.

Respectfully submitted,

Robert E. Malm
Reg. No. 34,662

Date: 11/09/05
Telephone: (310) 573-1781

Enc: Table of Contents (2 pages)
Grounds of Rejection to be Reviewed on Appeal (2 pages)
Summary of Claimed Subject Matter (10 pages)

TABLE OF CONTENTS

SUMMARY OF CLAIMED SUBJECT MATTER

SUMMARY OF CLAIMED SUBJECT MATTER FOR INDEPENDENT CLAIM 1

The invention is a method for repetitively executing a plurality of software packages at a plurality of rates utilizing a common set of computational resources. The method consists of counting contiguous time increments and executing a plurality of software packages. Specification: p. 4, lines 12-21. Each software package is executed during each time increment in one or more sequences of time increments. The time increments in each sequence recur at a predetermined rate, and the time increments assigned to one software package do not overlap the time increments assigned to any other of the plurality of software packages. Specification: p. 4, lines 5-11; p. 10, lines 18-22.

SUMMARY OF CLAIMED SUBJECT MATTER FOR INDEPENDENT CLAIM 26

The invention is an apparatus for repetitively executing a plurality of software packages at a plurality of rates utilizing a common set of computational resources. The apparatus counts contiguous time increments and executes a plurality of software packages. Specification: p. 4, lines 12-21. The apparatus executes each software package during each time increment in one or more sequences of time increments. The time increments in each sequence recur at a predetermined rate, and the time increments assigned to one software package do not overlap the time increments assigned to any other of the plurality of software packages. Specification: p. 4, lines 5-11; p. 10, lines 18-22.

MEANS-PLUS-FUNCTIONS & STEP-PLUS-FUNCTIONS

1. A method for repetitively executing a plurality of software packages at one or more rates, utilizing a common set of computational resources, the method comprising the steps:

[step plus function] generating a sequence of time intervals for each of the plurality of software packages, the time intervals belonging to one software package not overlapping the time intervals belonging to any other of the plurality of software packages (Specification: p. 4, lines 5-11; p. 10, lines 18-22);

[step plus function] executing a plurality of software packages, each software package being executed during the time intervals of its sequence of time intervals (Specification: p. 4, lines 12-21).

2. The method of claim 1 **[further limitation of step plus function]** wherein the plurality of software packages of the "executing" step includes only valid software packages (Specification: p. 11, lines 7-9), the method further comprising the step:

[step plus function] utilizing one or more tests to identify the software packages that are valid (Specification: p. 11, lines 9-10).

3. The method of claim 2 **[further limitation of step plus function]** wherein one of the tests for validity is a one's complement checksum test of a software package's program memory (Specification: p. 11, lines 10-11).

4. The method of claim 2 **[further limitation of step plus function]** wherein a software package is assigned its own dedicated memory region, one of the tests for validity being whether the address returned for the software package's initialization procedure lies within its dedicated memory region (Specification: p. 11, lines 11-12).

5. The method of claim 4 **[further limitation of step plus function]** wherein one of the tests is whether the address is returned within a predetermined time (Specification: p. 11, lines 15-17).

6. The method of claim 2 **[further limitation of step plus function]** wherein a software package is assigned its own dedicated memory region, the software package's dedicated memory region including a stack memory region and/or a heap memory region (Specification: p. 8, lines 5-7), one of the tests for validity being whether the stack memory range and/or the heap memory range assigned during the execution of the software package's initialization procedure and the various associated entry points lies within the software package's dedicated memory region (Specification: p. 11, lines 12-15).

7. The method of claim 6 **[further limitation of step plus function]** wherein one of the tests is whether the stack memory range and/or the heap memory range and the various associated entry points are returned within a predetermined time (Specification: p. 11, lines 15-17).

8. The method of claim 1 **[further limitation of step plus function]** wherein a software package is assigned its own dedicated memory region (Specification: p. 8, lines 5-7).

9. The method of claim 8 **[further limitation of step plus function]** wherein the software package's dedicated memory region includes a stack memory region, a software package's stack residing in the software package's stack memory region (Specification: p. 8, lines 5-7).

10. The method of claim 1 **[further limitation of step plus function]** wherein a software package includes background tasks as well as foreground tasks, the background tasks being performed after the foreground tasks have been completed (Specification: p. 12, lines 8-14).

11. The method of claim 10 **[further limitation of step plus function]** wherein a background task is an infinite loop (Specification: p. 12, lines 10-12).

12. The method of claim 10 **[further limitation of step plus function]** wherein the software package causes the power utilized in executing the software package to be minimized after completion of the background tasks (Specification: p. 12, lines 12-14).

13. The method of claim 1 **[further limitation of step plus function]** wherein a failure in the execution of a software package causes information to be logged in a failure log (Specification: p. 12, lines 15-18).

14. The method of claim 13 **[further limitation of step plus function]** wherein a failure in execution is linked to the software package that caused the failure (Specification: p. 12, lines 15-16).

15. The method of claim 13 **[further limitation of step plus function]** wherein quality of performance in executing a software package is represented by one or more performance-quality parameters, values of the one or more performance-quality parameters being determined from the information logged in a failure log, the execution of a software package being subject to a plurality of execution options, an execution option being selected on the basis of one or more performance-quality parameter values (Specification: p. 12, lines 15-22).

16. The method of claim 15 **[further limitation of step plus function]** wherein the plurality of execution options are user configurable (Specification: p. 12, lines 21-22).

17. The method of claim 15 **[further limitation of step plus function]** wherein performance-quality parameters include the number of failures and/or the rate of failures for one or more classes of failures recorded in a software package's failure log (Specification: p. 12, lines 19-20).

18. The method of claim 1 **[further limitation of step plus function]** wherein safety-critical software is placed in one or more separate partitions thereby isolating the safety-critical software from non-safety-critical software (Specification: p. 13, lines 1-2).

19. The method of claim 1 **[further limitation of step plus function]** wherein each of the plurality of software packages is assigned its own memory block, a software package being enableable to read data only from zero or more memory blocks associated with other software packages, the zero or more memory blocks readable by a software package being either predetermined or determined during execution of the software packages in accordance with a set of one or more rules (Specification: p. 8, line 20 - p. 9, line 14).

20. The method of claim 1 **[further limitation of step plus function]** wherein each of the plurality of software packages is assigned its own memory block, a software package being enableable to write data only to zero or more memory blocks associated with other software packages, the zero or more memory blocks writeable by a software package being either predetermined or determined during execution of the software packages in accordance with a set of one or more rules (Specification: p. 8, lines 20 - p. 9, line 6 and lines 15-18).

21. The method of claim 1 **[further limitation of step plus function]** wherein an executive software package enforces the discipline that each software package executes only during the time intervals of its sequence of time intervals, the executive software package determining when the execution of a software package extends into a time interval belonging to the sequence of time intervals assigned to another software package and performs a remedial action (Specification: p. 10, line 18 - p. 11, line 6).

22. The method of claim 1 **[further limitation of step plus function]** wherein the presence of those software packages that are present is detected (Specification: p. 11, lines 7-18).

23. The method of claim 1 **[further limitation of step plus function]** wherein one or more of the plurality of software packages are independently compiled, linked, and loaded (Specification: p. 10, lines 7-8).

24. The method of claim 1 **[further limitation of step plus function]** wherein a software package has its own stack, the software package's stack being selected prior to executing the software package (Specification: p. 11, lines 19-22).

25. **[means plus function]** Apparatus for practicing the method of claim 1 (Specification: p. 4, lines 5-21; p. 10, lines 18-22).

26. Apparatus for repetitively executing a plurality of software packages at a plurality of rates, the apparatus comprising:

[means plus function] a means for generating a sequence of time intervals for each of the plurality of software packages, the time intervals belonging to one software package not overlapping the time intervals belonging to any other of the plurality of software packages (Specification: p. 4, lines 5-11; p. 10, lines 18-22);

[means plus function] a means for executing a plurality of software packages, each software package being executed during the time intervals of its sequence of time intervals (Specification: p. 4, lines 12-21).

27. The apparatus of claim 26 **[further limitation of means plus function]** wherein the plurality of software packages executed by the "executing" means includes only valid software packages (Specification: p. 11, lines 7-9), the apparatus further comprising:

[means plus function] a means for utilizing one or more tests to identify the software packages that are valid (Specification: p. 11, lines 9-10).

28. The apparatus of claim 27 **[further limitation of means plus function]** wherein one of the tests for validity is a one's complement checksum test of a software package's program memory (Specification: p. 11, lines 10-11).

29. The apparatus of claim 27 **[further limitation of means plus function]** wherein a software package is assigned its own dedicated memory region, one of the tests for validity being whether the address returned for the software package's initialization procedure lies within its dedicated memory region (Specification: p. 11, lines 11-12).

30. The apparatus of claim 29 **[further limitation of means plus function]** wherein one of the tests is whether the address is returned within a predetermined time (Specification: p. 11, lines 15-17).

31. The apparatus of claim 27 **[further limitation of means plus function]** wherein a software package is assigned its own dedicated memory region, the software package's dedicated memory region including a stack memory region and/or a heap memory region (Specification: p. 8, lines 5-7), one of the tests for validity being whether the stack memory range and/or the heap memory range assigned during the execution of the software package's initialization procedure and the various associated entry points lies within the software package's dedicated memory region (Specification: p. 11, lines 12-15).

32. The apparatus of claim 31 **[further limitation of means plus function]** wherein one of the tests is whether the stack memory range and/or the heap memory range and the various associated entry points are returned within a predetermined time (Specification: p. 11, lines 15-17).

33. The apparatus of claim 26 **[further limitation of means plus function]** wherein a software package is assigned its own dedicated memory region (Specification: p. 8, lines 5-7).

34. The apparatus of claim 33 **[further limitation of means plus function]** wherein the software package's dedicated memory region includes a stack memory region, a software package's stack residing in the software package's stack memory region (Specification: p. 8, lines 5-7).

35. The apparatus of claim 26 **[further limitation of means plus function]** wherein a software package includes background tasks as well as foreground tasks, the background tasks being performed after the foreground tasks have been completed (Specification: p. 12, lines 8-14).

36. The apparatus of claim 35 **[further limitation of means plus function]** wherein a background task is an infinite loop (Specification: p. 12, lines 10-12).

37. The apparatus of claim 35 **[further limitation of means plus function]** wherein the software package causes the power utilized in executing the software package to be minimized after completion of the background tasks (Specification: p. 12, lines 12-14).

38. The apparatus of claim 26 **[further limitation of means plus function]** wherein a failure in the execution of a software package causes information to be logged in a failure log (Specification: p. 12, lines 15-18).

39. The apparatus of claim 38 **[further limitation of means plus function]** wherein a failure in execution is linked to the software package that caused the failure (Specification: p. 12, lines 15-16).

40. The apparatus of claim 38 **[further limitation of means plus function]** wherein quality of performance in executing a software package is represented by one or more performance-quality parameters, values of the one or more performance-quality parameters being determined from the information logged in a failure log, the execution of a software package being subject to a plurality of execution options, an execution option being selected on the basis of one or more performance-quality parameter values (Specification: p. 12, lines 15-22).

41. The apparatus of claim 40 **[further limitation of means plus function]** wherein the plurality of execution options are user configurable (Specification: p. 12, lines 21-22).

42. The apparatus of claim 40 **[further limitation of means plus function]** wherein performance-quality parameters include the number of failures and/or the rate of failures for one or more classes of failures recorded in a software package's failure log (Specification: p. 12, lines 19-20).

43. The apparatus of claim 26 **[further limitation of means plus function]** wherein safety-critical software is placed in one or more separate partitions thereby isolating the safety-critical software from non-safety-critical software (Specification: p. 13, lines 1-2).

44. The apparatus of claim 26 **[further limitation of means plus function]** wherein each of the plurality of software packages is assigned its own memory block, a software package being enableable to read data only from zero or more memory blocks associated with other software packages, the zero or more memory blocks readable by a software package being either predetermined or determined during execution of the software packages in accordance with a set of one or more rules (Specification: p. 8, line 20 - p. 9, line 14).

45. The apparatus of claim 26 **[further limitation of means plus function]** wherein each of the plurality of software packages is assigned its own memory block, a software package being enableable to write data only to zero or more memory blocks associated with other software packages, the zero or more memory blocks writeable by a software package being either predetermined or determined during execution of the software packages in accordance with a set of one or more rules (Specification: p. 8, line 20 - p. 9, line 6 and lines 15-18).

46. The apparatus of claim 26 **[further limitation of means plus function]** wherein an executive software package enforces the discipline that each software package executes only during

the time intervals of its sequence of time intervals, the executive software package determining when the execution of a software package extends into a time interval belonging to the sequence of time intervals assigned to another software package and performs a remedial action (Specification: p. 10, line 18 - p. 11, line 6).

47. The apparatus of claim 26 **[further limitation of means plus function]** wherein the presence of those software packages that are present is detected (Specification: p. 11, lines 7-18).

48. The apparatus of claim 26 **[further limitation of means plus function]** wherein one or more of the plurality of software packages are independently compiled, linked, and loaded (Specification: p. 10, lines 7-8).

49. The apparatus of claim 26 **[further limitation of means plus function]** wherein a software package has its own stack, the software package's stack being selected prior to executing the software package (Specification: p. 11, lines 19-22).

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

I. WHETHER CLAIMS 1, 25, AND 26 ARE UNPATENTABLE UNDER NON-STATUTORY DOUBLE-PATENTING DOCTRINE IN VIEW OF MARK ET AL. (U.S. 6,829,763).

II. WHETHER CLAIMS 1-49 ARE UNPATENTABLE UNDER 35 U.S.C. § 112, SECOND PARAGRAPH, AS BEING INDEFINITE.

III. WHETHER CLAIM 25 IS UNPATENTABLE UNDER 35 U.S.C. § 112, SECOND PARAGRAPH, FOR FAILING TO SET FORTH THE SUBJECT MATTER WHICH APPLICANTS REGARD AS THEIR INVENTION.

IV. WHETHER CLAIM 25 IS UNPATENTABLE UNDER 35 U.S.C. § 101 BECAUSE IT EMBRACES OR OVERLAPS TWO DIFFERENT STATUTORY CLASSES.

V. WHETHER CLAIMS 1, 8, 23, 25, 26, 33, AND 48 ARE UNPATENTABLE UNDER 35 U.S.C. § 102(B) AS BEING ANTICIPATED BY BLUM ET AL. (U.S. 4,109,311).

VI. WHETHER CLAIMS 10-11, 13-14, 18, 35-36, 38-39, AND 43 ARE UNPATENTABLE UNDER 35 U.S.C. § 103(A) IN VIEW OF BLUM ET AL. (U.S. 4,109,311).

GROUPING OF CLAIMS

Insofar as Ground I is concerned:

Claims 1, 25, and 26 stand or fall together;

Insofar as Ground II is concerned:

Claims 1 and 26 stand or fall together;

Claims 21 and 46 stand or fall together;

Claims 23 and 48 stand or fall together;

Claims 24 and 49 stand or fall together;

Claims 2 and 27 stand or fall together;

Claims 3 and 28 stand or fall together;

Claims 6 and 31 stand or fall together and claims 7 and 32 stand or fall together;

Claims 18 and 43 stand or fall together;

Claims 22 and 47 stand or fall together;

All other claims stand or fall with the claims from which they depend.

Insofar as Ground III is concerned:

Claim 25 stands or falls by itself.

Insofar as Ground IV is concerned:

Claim 25 stands or falls by itself.

Insofar as Ground V is concerned:

Each of claims 1, 8, 23, 25, 26, 33, and 48 stands or falls by itself.

Insofar as Ground VI is concerned:

Each of claims 10-11, 13-14, 18, 35-36, 38-39, and 43 stands or falls by itself.